Available online at www.sciencedirect.com
ScienceDirect
Nuclear and Particle Physics Proceedings 273–275 (2016) 138–147
www.elsevier.com/locate/nppp

Computing in high-energy physics

Richard P. Mount

SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025, USA

Abstract

I present a very personalized journey through more than three decades of computing for experimental high-energy physics, pointing out the enduring lessons that I learned. This is followed by a vision of how the computing environment will evolve in the coming ten years and the technical challenges that this will bring. I then address the scale and cost of high-energy physics software and examine the many current and future challenges, particularly those of management, funding and software-lifecycle management. Finally, I describe recent developments aimed at improving the overall coherence of high-energy physics software.

Keywords: Computing; Technology evolution; Software; Software lifecycle

1. A personal view of four decades of computing

The personalized approach ensures that my account is rigidly founded on reality, albeit on a limited reality. It might also be hoped that it contributes readability that offsets the inevitable distortion with respect to wider truth.

1.1. In the beginning

As I was leaving high school (or in the language of the place and time, “grammar school”), already knowing that I would study physics at Oxford University, I accepted the farsighted advice of my Oxford tutor to “find out something about computers”. I got a summer job at a computer services start-up in a manufacturing town in Yorkshire. I learned to program fast in bad COBOL, fast and accurate punching on a totally manual 12-key punch, and networking (using Ford van technology).

As an undergraduate, apart from running some least squares fits to laboratory experiment data on a PDP-8, I didn’t encounter computers, but as a Cambridge University graduate student I was immediately surrounded by the computing paraphernalia of bubble-chamber physics. I visited CERN and joined in taking hundreds of thousands of pictures of the 2-metre hydrogen bubble chamber which I analysed with the aid of a human trigger (scanners), a PDP-8/S data acquisition system, bicycle-powered networking, and analysis via opportunistic use of the astronomer’s IBM 360/44. I learned FORTRAN and OS/360 JCL. I also found out that people (scanners) work better when motivated, and that high-energy physics (HEP) already involved a lot of arcane software.

1.2. The European Muon Collaboration (EMC)

My experiences as a postdoc on the EMC experiment were to shape my view of the role of computing. EMC, with 99 physicists, was the largest collaboration of its day, building and then exploiting a massive and complex detector at the end of a purpose-built 2-km muon beamline at CERN. As a

http://dx.doi.org/10.1016/j.nuclphysbps.2015.09.017

flagship experiment of its day, EMC ran for a large fraction of each year writing raw-data tapes at a rate of up to one tape every ten minutes. The resultant 10,000 tapes per year is numerically comparable to the tape use of an LHC experiment.

In spite of this expected data rate, computing resource planning could be summarised as “we hope we can process the data somewhere.” As a result, even first-pass reconstruction was often delayed by over a year, major detector deterioration was not discovered until dangerously late, and physics analysis had to fight large systematic effects. I saw responsible physicists trying for almost two years to ascribe an unexpected physics result [1] to systematics, before finally giving in and publishing.

One of my personal contributions to EMC was to invent the technique for beam normalisation (luminosity measurement). I proposed exploiting the installed beam-muon tracking system in a novel way [2] that required randomly triggering during the beam burst and writing out large numbers of very small events. “Can’t do that, you will fill up the tapes with inter-record gaps” was the response of my colleagues. So I implemented a new I/O layer on a number of architectures and developed and operated a 10,000 tape/year data-management system that separated EMC data into physics and trigger streams, including, of course, my own luminosity trigger.

By the time I moved on from EMC, I was incurably convinced that computing resource planning was essential, that data management mattered, and that software quality mattered even more than software efficiency. The latter conviction came from fighting to understand code that had been shoehorned into the CDC 7600’s Small Core Memory by heavy re-use of variables within FORTRAN.

In many ways, EMC set the course of my future by offering graphic and often painful demonstrations that software and computing matter in HEP.

1.3. L3 at CERN’s LEP acclerator

After EMC, I joined Harvey Newman in planning computing for the L3 experiment and in creating, right down to making cables, the “LEP3NET” US-CERN network that is the ancestor of today’s transatlantic component of LHCOPN[1]. Our 1983 computing planning, six years ahead of LEP start-up, estimated CPU that turned out to be low by a factor of 1000 and planned to use disk only for tape staging. However, the cost estimate was approximately correct! Criticism from CERN management that our requests were irresponsibly greedy was taken as some measure of validation, especially as we were not expecting CERN to provide most of the resources.

I learned that, not only in war, *“plans are worthless, but planning is everything”* [3]. I also learned that “what are our requirements” is the wrong question in computing for HEP. The right question is something like “what will affordable technology be able to do for our physics productivity,” where technology includes CPU, disk, tape and networks.

2. BaBar

BaBar was close to being the entirety of the SLAC HEP program in 1997 when I joined the laboratory to take charge of computing. I found a laboratory totally committed to the success of BaBar, but struggling to plan for BaBar computing. The 1995 Technical Design Report called for 17,500 MIPS of CPU, 5 TB of disk and planned all data movement between centres to be on tape. Data taking started in 1999 and within two years BaBar had 1,700,000 MIPS, 80 TB of disk, and had moved data around the world using only the network. Fortunately SLAC management had guessed that huge resources would be needed and I only had to confirm and quantify this viewpoint.

While the resource-planning experience served as a good endorsement of the Eisenhower quotation, BaBar also provided an experience from which the lessons are harder to extract. In 1997, with some encouragement from me, BaBar took the bold step of choosing an object database management system, Obectivity/DB, to manage the storage of, and access to, its event data. BaBar’s Objectivity database was scaled to close to 1 PB between 1999 and 2003, when it was abandoned with much rejoicing on the part of the BaBar physicists. An analysis of this experience would fill more space than I am allowed, so I will reduce the message to my personal distillation: making strategic software decisions is hard!

I will avoid the temptation to attempt to write wise words about my next adventure, computing for the ATLAS LHC experiement, and move almost immediately to an attempt to look into the future, somewhat informed by past experience.

[1] The LHC Optical Private Network

3. The next ten years of technology evolution

3.1. Cautionary Tales

First, I must write some cautionary words about the clarity of my crystal ball. In 1986, in lectures at the CERN School of Computing, I argued that Seymour Cray's CDC 7600, released seventeen years earlier, was still a fast machine [4]. This was factually correct, but I drew the inference that CPU evolution had stalled and devoted much effort to parallel computing about 25 years earlier than necessary.

It gets better! In 1991, when I had major computing responsibilities in L3, two guys I barely knew, Tim and Robert [5], presented a weird system of distributed hypertext to me. I thought that L3, and the world, could do just fine without this.

By 1996 I had understood that for any technology, relevance to the future of HEP was much more linked to the technology's future market success than to its exact match to our needs. Accordingly, my recommendation to BaBar to embark on the Objectivity adventure was based on my guess that the market for object database management systems would grow rapidly to challenge the market for relational database management systems. It turned out that I was optimistic by at least three orders of magnitude.

3.2. A Narrow View of History

I have been buying computing technology for HEP for more than three decades, giving me a very narrow, but arguably very relevant, view of the progress of the computing technologies we use.

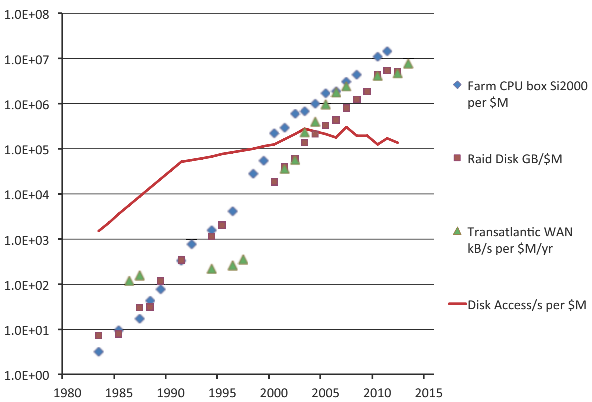

Figure 1. CPU, disk and transatlantic networking purchased by R. Mount and H. Newman as a function of time. For each purchase, what was or could be bought for $1 million is shown.

Figure 1 shows a summary of what I could buy for one million dollars as a function of time. Harvey Newman of Caltech and I were intimately involved in HEP's transatlantic networking in the period up to 1997. Since then I have been a user not a provider and the later networking points in the figure are based on information from Harvey.

The points for CPU and disk technology per $ are quite good fits to straight lines on the logarithmic plot with slopes corresponding to doubling in less than 18 months, over a period of 30 years. The wide area network points have a more complex behaviour, reflecting the highly regulated environment in the first 15 years that became more market-driven in subsequent years but with a large inertia related to the cycle of installing undersea cables.

The line showing disk accesses/s per $M relates to random access to small amounts of data. The 20-year stagnation is an obvious consequence of the largely unchanging rotational speed of disks. Before the early 1990s, disks were too expensive to be used to store HEP data. With the introduction of the first 1GB 3.5 inch disk [6] by IBM in 1991, disks supplanted tapes as the store for HEP "DSTs" (literally Data Summary *Tapes*) used in physics analysis, and could deliver sparsely accessed events faster than they could be analysed. Twenty years later, the ratio of disk accesses to CPU power had degraded by over four orders of magnitude and the only way to use disks for analysis data was to, once again, access data serially as if they were on tape.

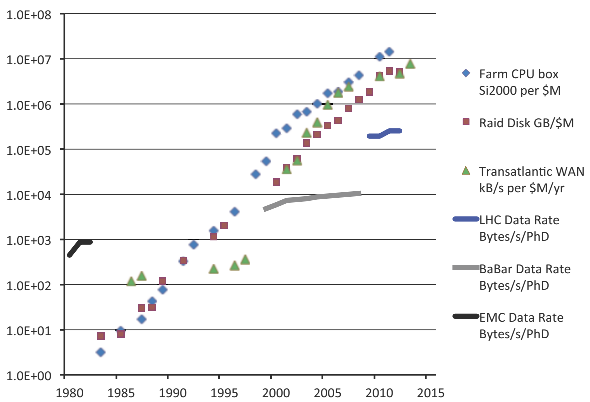

Figure 2. The CPU, disk and networking points from Figure 1 in comparison with the data acquisition rates written to persistent storage of EMC, BaBar and the two general-purpose LHC experiments. The experiment's data acquisition rates are expressed as bytes per second per PhD physicist.

Figure 2 shows the CPU, disk and network points in comparison with the data rates of EMC, BaBar and the general-purpose LHC detectors. Based on the thesis that funding per experiment varies widely, but funding per PhD physicist involved in an experiment varies much less, the experiments' data rates are normalized to the number of participating PhD physicists. The relationship of the data-rate lines to the technology points gives a coarse indication of how hard it is or was to analyse the experiment's data with the available computing technologies and funding. Clearly EMC was close to impossible and BaBar was difficult whereas the LHC experiments are slightly easier!

3.3. Predictions

In the BaBar era, I was responsible for the model that predicted the funding needed for computing in future years. I was able to use simple extrapolations from my experience, for example doubling in technology per $ every 18 months, and I was not proved wrong. As we all know, that approach no longer works very well.

3.4. CPU Predictions

The steady increase of processor clock speeds, that drove much of the CPU evolution for decades, ended in about 2005. However Moore's Law [7] was not repealed, and the exponential growth of transistors was used to make two or more compute cores per processor and to add complexity and function to each core. The result was that no knee is visible in the CPU points in Figure 1, largely due to experimental HEP's ability to use trivial event-level parallelism to exploit multi-core processors. Intel is expected[2] to continue to shrink the feature size in its processors, leading to continued exponential growth in the number of transistors, for the next ten years. Figure 3 shows the expected die shrinks (ticks) and microarchitecture improvements (tocks) leading to a factor 19 more transistors per unit area by 2021 assuming one tick or tock per year, or by 2024, with a more conservative prediction of a tick or tock every 18 months. Can HEP use all these transistors? Clearly the answer is "no", because we already use existing Intel CPUs quite poorly. Even in 2010, it was already clear that in HEP event processing "the CPU is not doing anything useful during 70% - 95% of the time" [8]. My personal guess is that with a large effort to develop highly multithreaded, memory-parsimonious code, our misuse of the hardware will not get worse as the number of transistors increases. HEP has been particularly ineffective in using the increasingly capable vector-processing hardware within modern processors and there is hope, that I share to a small extent, that this situation can be alleviated by changes to the execution architecture of our code.

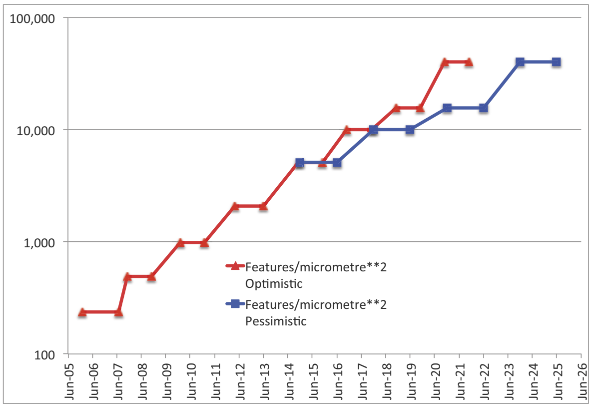

Figure 3. Historical and predicted progress of Intel's "tick-tock" feature-size shrinkage alternating with microarchitecture improvements. The predictions are taken from Wikipedia.

3.5. Disk Predictions

Disk technology is in trouble. Quoting from one irreverent technology monitoring publication [9]: "The future of storage: disk-based or just discombobulated? You want disk tech innovations? We got 'em, lots of 'em." My own limited sampling of expert opinion within close connections to the disk industry concurs, even if in more measured language. The problem is that "feature sizes" on current magnetic disk surfaces have reached the limit of magnetic stability. There are indeed "lots" of ways to overcome this limitation and produce higher capacity devices. The problem is that all the technologies are immature and there is no prediction of the final market winner.

To get more data on to magnetic disk surfaces, the industry must either use higher coercivity material, or must maintain the volume of a recorded bit as its surface area shrinks. "Heat-assisted magnetic recording" (HAMR) is typical of the first approach and "shingled recording" is typical of the second. Each comes with its technological issues. My conclusion is that disk dollars per byte will progress

[2] wikipedia.org/wiki/Intel_Tick-Tock [quoting Wikipedia guesses]

more slowly in the next ten years than any other technology in HEP computing.

Discussion of the future of disk technology would not be complete without mentioning solid-state disks (SSDs), which are, of course, not disks at all. SSDs have already replaced rotating disks in most laptops, but the global flash-memory fabrication capability is far too small to challenge rotating disks for bulk data storage, and there is no expectation that it could do so at attractive prices. If rotating disks remain permanently in the doldrums, SSDs might eventually dominate, but at prices that would be very unwelcome to the HEP community.

In one respect SSDs offer vastly better value for money than rotating disks. Had I plotted the SSD values for accesses per second per million dollars in Figure 1, it would have shown that, where random access is important, SSDs are much better value for money.

3.6. Tape predictions

Tape also did not appear in Figure 1. Quite simply I only bought tape systems in the BaBar years and turning the sporadic expenditures on robots, drives and media into a meaningful set of points proved beyond me. However, much to the delight of those who rejoice in the apparently absurd, tape is again rising in importance due, of course, to the dismal state of disk technology. Tape systems cost about 1/3 the price per byte of disk systems and offer greater data integrity. The tape market is still relatively small, and has never benefitted or suffered from the imperative to bring the highest performing technology rapidly to market to compete successfully. Tape magnetic "feature sizes" are far from physical limits and the industry consortium INSIC[3] predicts that areal densities will double every two years, leaving disk in the dust. My personal estimate is that this is at the upper end of what can be expected.

3.7. Network predictions

As in Figure 1, I continue to focus on wide-area networks. Here cost evolution is much more dictated by market volume and long lead times for undersea cables rather than technology, since protectionist regulation has largely disappeared. One international survey [10] reported 49% annual growth in demand between 2008 and 2012, and I consider that recent history is probably the best guide to future evolution.

[3] Information Storage Industry Consortium, www.insic.org/

Table 1. Personal expectations for the relative improvement in technology costs between 2014 and 2024

Technology per $	Factor
CPU transistors	10 to 32
Disk capacity	4 to 8
Tape capacity	8 to 32
WAN bandwidth	10 to 58

3.8. Prediction summaries and likely impact on HEP

Table 1 summarises my predictions. The geometric means of these factors are used to draw the technology extrapolation lines in Figure 4, which extends Figure 2 by ten years. Also added are my expectations for Run 3 and Run 4 data-acquisition rates for the general-purpose LHC experiments. With respect to LHC, I draw the tentative conclusion that disk space will be a problem, but that CPU will be no more limiting than at present, provided we undertake the massive task of making our code highly multithreaded.

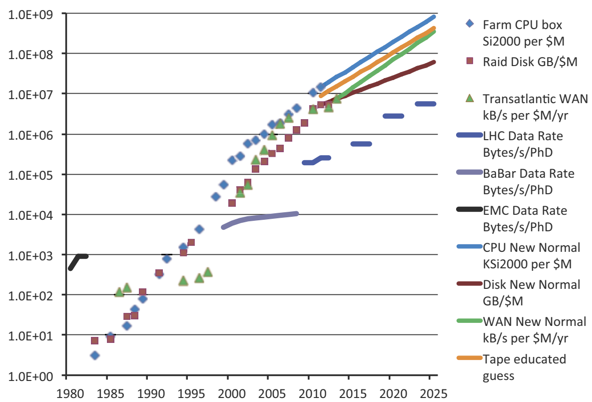

Figure 4. Personal estimates of the likely evolution of CPU, disk, tape and WAN technology shown in relation to the data in Figure 2. My estimates of LHC Run 3 and Run 4 data acquisition rates are also shown.

Figure 4 suggests that there will be a significant increase in the cost of disk relative to other elements of the HEP computing environment. This also suggests the three elements of the way forward:

1. Exploit CPU by re-computing more derived data and storing less;

2. Exploit tape to store less frequently needed data;
3. Exploit the WAN to move data rapidly when needed or to access data remotely.

Could HEP automate all the necessary decision making? I believe that after identifying key physics policy requirements, such as the data-integrity requirements and the lifetime of the various types of derived data, we should be able to leave all other decisions to “the system”.

4. Software

I begin with a quote [11] from the late David Williams:

- Until the mid 1980s HEP’s “computing problem” was often thought to be about obtaining enough CPU power;
- Then we worried about storage capacity;
- The real problem has always been, in my opinion, getting people to collaborate on a solution.

I will return to this theme towards the end of this section on software.

As for the hardware story, I will centre my examination of software on personal experience, particularly looking at questions of cost and organization rather than actually writing software.

4.1. HEP software, scales and costs

I choose to examine Root [12], xrootd [13] and Geant4 [14]. I have played a “godfather[4]” role in Geant4, and a godfather-cum-line-manager[5] role in xrootd. I am presenting an analysis of cost that I first saw on the Root website, so the inclusion of Root seemed natural.

Figure 5 shows a simple cost analysis from Open Hub[6] of the Root code. This analysis finds 1.75 million lines of code and infers an expenditure of 505 person-years and a cost of \$27.6 million, making an salary assumption that is only a fraction of the real cost of the Root developers who are mainly CERN staff.

The same analysis for xrootd finds 150k lines of code, 28 person-years and a cost of \$2.1 million. In this case, I can approximately confirm the effort estimate. About 50% of xrootd development took place at SLAC, a high-cost environment where I get no change from \$1 million if I employ three scientists for a year[7]. The remainder took place at a combination of lower-cost institutions and CERN. The Open Hub estimate of cost is probably low by about a factor of four.

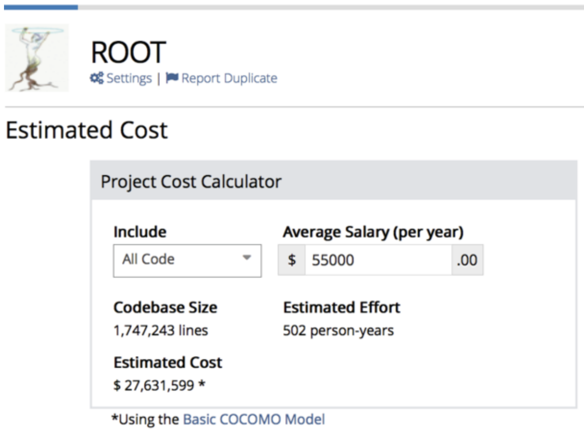

Figure 5. The Open Hub analysis of the effort and cost required to create the contents of the Root software repository

Geant4, my final example, tips the scales at 2.1 million lines of code and an estimated 602 person-years. My own guess would be 25 full-time equivalents for 20 years, so slightly less. The Open Hub cost estimate is \$33.1 million. I suspect that this is also low, but Geant4 effort comes from a wide

[4] I was one of the two referees for the original Geant4 project in 1994. I hired the SLAC Geant4 team in the late 1990s and was Geant4 Collaboration Board chair for six years. I and am now the US representative on the Geant4 Oversight Board.

[5] Xrootd emerged from the server technology created by A. Hanushevsky at SLAC to make Objectivity/DB scale to meet BaBar needs. When Objectivity was dumped, the technology was re-implemented to make Root data access scale to Petabytes. For most of its life, I have been responsible for supporting and seeking funding for xrootd at SLAC.

[6] This and many other analyses are performed by Black Duck Open Hub, known formerly as Ohloh. The URL at the time of retrieval was https://www.openhub.net/p/ROOT/estimated_cost

[7] SLAC, like many US institutes no longer has a “base budget”. Thus all funds brought in to the laboratory must contribute their share of the cost of the roads, the buildings, the management, the purchasing department, the human resources department, the safety and health department etc. The resultant “fully burdened” cost appears to bear little relation to the salary of the scientist, but is a fair estimate of the cost of the work to the taxpayer.

variety of institutes, many of which do not think, like SLAC, in terms of fully burdened cost.

These three examples of software important to HEP are estimated by Open Hub to have cost us about $63 million for four million lines of code. I am sure this is too low by at least a factor of two. These examples also represent a very small fraction of HEP software. The BaBar code repository amounts to over 9 million lines of code, therefore costing over $150 million. The LHC experiments have outdone this achievement comfortably and have software costs in the hundreds of millions ($, Euros or Swiss Francs as you wish).

All the examples above could be regarded as relating to event-processing software. The story of the HEP software enabling distributed (also known as Grid) computing is also important. This will be addressed below.

4.2. The fascinating story of the Grid

The Grid, in both the etymological and initial software sense was invented by Foster and Kesselman in the late 1990s [15]. My Grid adventure started in 1998, with an opportunity to apply for non-HEP "Next Generation Internet" funding from the US Department of Energy. Mindful of the funding value of the "G" word, together with Harvey Newman I assembled a collaboration of HEP enthusiasts and real computer scientists and submitted the proposal for the "Particle Physics Data Grid (PPDG)." The list of collaborators shown in Figure 6 is interesting in itself. There are 31 names, almost equally divided between physicists, academic computer scientists and HEP computing specialists.

California Institute of Technology	**Harvey B. Newman**, Julian J. Bunn, James C.T. Pool, Roy Williams
Argonne National Laboratory	**Ian Foster**, Steven Tuecke **Lawrence Price**, David Malon, Ed May
Berkeley Laboratory	**Stewart C. Loken**, Ian Hinchcliffe **Arie Shoshani**, Luis Bernardo, Henrik Nordberg
Brookhaven National Laboratory	**Bruce Gibbard**, Michael Bardash, Torre Wenaus
Fermi National Laboratory	**Victoria White**, Philip Demar, Donald Petravick **Matthias Kasemann**, Ruth Pordes
San Diego Supercomputer Center	**Margaret Simmons**, Reagan Moore,
Stanford Linear Accelerator Center	**Richard P. Mount**, Les Cottrell, Andrew Hanushevsky, David Millsom
Thomas Jefferson National Accelerator Facility	**Chip Watson**, Ian Bird
University of Wisconsin	**Miron Livny**

Figure 6. The names on the 1998 Particle Physics Data Grid proposal

The PPDG proposal abstract began "The Particle Physics Data Grid has two objectives: delivery of an infrastructure for widely distributed analysis of particle physics data at multi-petabyte scales by thousands of physicists, and acceleration of the development of network and middleware infrastructure aimed broadly at data-intensive collaborative science." We proposed to deliver all of this with 22 person-years at a total cost of $4 million. As I was writing "$4 million" with one half of my brain, the other half was comparing the complexity of our goals with that of, for example, a new release of Microsoft Windows and secretly guessing that a cost of $300 million might be slightly more accurate. The goal stated in the abstract was spot on, but what we could achieve with our small project was a small fraction of the goal. One achievement that we underestimated when making the proposal was that of establishing communication and collaboration between physicists and computer scientists. This proved to be challenging, amusing, and ultimately very rewarding.

From my perspective, PPDG was the first of the HEP Grid projects, but it was by no means the last. Figure 7 shows a simplified timeline of the HEP-relevant Grid projects on both sides of the Atlantic.

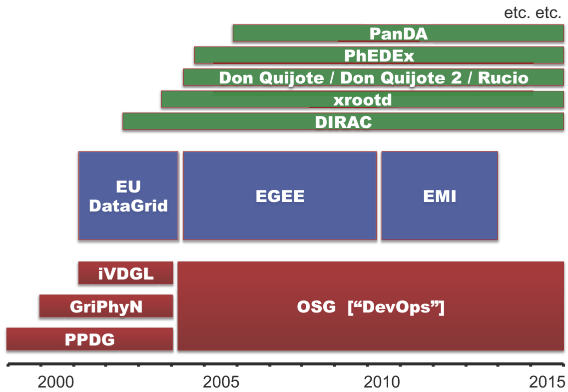

Figure 7. Timeline of Grid projects in the USA (lower bars) and Europe (middle bars) together with more experiment-specific software supporting distributed computing

The US and EU Grid projects have cost, in total, more than $100 million to date for software development and some operational support. Noting my $300 million guesstimate, it should come as no surprise that these projects did not deliver a complete distributed computing environment. The ultimate outstanding success (at least in my eyes) of LHC distributed computing required the development of a plethora of additional data and workload management systems, some of which are indicated in the upper third of Figure 7.

So what did all this cost, excluding all hardware and operational effort? Certainly the cost is well over \$100 million and seems well on its way towards my \$300 million guess. Did any duplicative or wastefully uncoordinated work happen in the 16 years? Of course it did, and it can now be clearly identified with hindsight. However, the support for the LHC physics program must be recognized as a great success. The support for science in general, or even for non-LHC HEP is far more questionable.

Taking as unassailable the funding imperatives that lead HEP to distribute data and computing around the world, when I ask myself "was our Grid adventure reasonably cost-effective?" I have to echo Deep Thought [16] and respond "tricky."

4.3. HEP software: can we collaborate more effectively?

In the fascinating story of the Grid, I noted explicitly and implicitly many areas of duplication of function and imperfect global coordination. This picture remains when we look at many other areas of HEP software. However, when undertaking something new, allowing competing approaches is usually valuable. We must also be realistic about the overheads of multi-experiment software collaboration.

I also noted the very limited benefit of our Grid projects to fields other than HEP. However, the HEP record is by no means uniformly bad – there are clear examples of collaboration on software of general value to the field and beyond. To my mind, the prime example is Geant4.

Figure 8 shows the geographic breadth of the Geant4 collaboration which started life as a research project in 1984 and has achieved a resounding success in meeting the needs of LHC physics, and in becoming the unique toolkit for detailed simulation in HEP. But beyond this, Geant4 has become a poster child for the potential usefulness of HEP software in other fields. The European Space Agency was an early contributor to Geant4 and the software is now widely used by ESA, NASA and their contractors to model crucial effects such as the effects of radiation on humans and on semiconductor devices.

Geant4 also has a rapidly growing applicability in medicine, both in radiobiology and in reliable oncology treatment planning.

So HEP as a whole should just copy the Geant4 Collaboration? That would be far too simplistic, but there are elements of the Geant4 experience that can inform efforts to optimize collaboration within HEP software - perhaps most notably a studious effort to politely decline top-down management of the whole enterprise by any single region or institute.

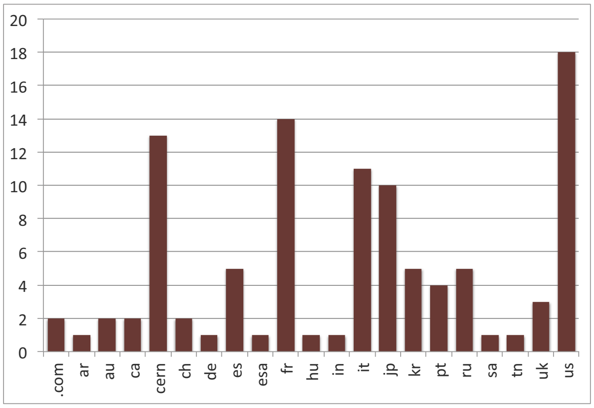

Figure 8. The affiliations of the 104 members of the Geant4 Collaboration. This distribution should not be taken too seriously – for example ESA, the European Space Agency, funds several activities at collaborating institutes but the members appear under their institutes rather than ESA.

Continuing the management theme we can ask "how has successful HEP software been managed?" Geant4 was a spontaneous community initiative, similar to the formation of an experimental collaboration. Like any successful HEP experiment, Geant4 enjoyed sustained resource-provider support, significantly encouraged, in my opinion, by the collaboration's adoption of a process to organize fairly regular peer reviews of its science, its software and its organization.

Another example of resounding success within HEP is Root. The Root team suffered years of management indifference or weak opposition. Only when Root was becoming indispensible to the CERN physics program did management acknowledge this and provide explicit resources.

This does not mean that management is stupid, but it does mean that the developers willing to risk their careers to contribute to a collaborative software project often have a slightly clearer view of the future than can be achieved by management.

An aspect that nobody seems to be particularly good at is managing the software lifecycle, from great idea, through prototype, version 1.0, longer-term maintenance to final decent burial. The problems begin with maintenance, which is not exactly career-enhancing, and become severe when any consideration of decent burial is appropriate. Making the decision to terminate support for cross-cutting software is even more challenging than setting up the collaboration to create it. Funding sources are keenly aware of this and rightly fear that cross-cutting software may generate “entitlements” to on-going funding that they have no idea how to control. Managing a portfolio of such software is really difficult. The HEP community needs to show that it can provide this management to be able to attract anything beyond short-term project funding.

It has to be said that, when software development is confined to an experiment, all aspects of management become simpler and more acceptable because organization and management are so clearly required to meet the experiment’s goals.

Before addressing how HEP might meet the challenges of optimally collaborative software development I look briefly at some of the technical and less-technical challenges that we face.

4.4. LHC software challenges

In discussing Figure 4, I indicated that technology evolution should, from a sufficiently detached viewpoint, will allow us to pursue a rich programme of physics at the High Luminosity LHC. This optimistic view quietly assumes that we do an enormous amount of work. Anything like our current code will be a disaster in 2024. The CPU industry will present us with too many transistors, dubiously valuable vector capability, insufficient on-chip cache and far too little memory to continue with event-level parallelism. In principle we know what to do about this – decompose our code into many independent threads that can be mapped on to a variety of hardware architectures – but the task is huge and there will be many unpleasant surprises along the way.

Beyond the CPU issues, storage, already the dominant cost in LHC computing, will become more and more of a constraint. We cannot buy our way out of this and have no choice but to embark on the creation of a dynamic distributed computing system that will optimize the use of CPUs, disk, tapes and networking.

We could almost survive without these huge software efforts if we tightened our physics focus and tightened our triggers. This would ensure that we ignore the truly unexpected and to me this is unacceptable.

4.5. Non-LHC software challenges

Gross inequality is dangerous for any community and is certainly damaging to the future of HEP. Smaller, but not necessarily small, experiments now look hungrily at the rich LHC software and distributed computing environment. They see little benefit from the LHC software successes. The LHC experiments are not funded to provide support to smaller experiments, and much of the hugely successful LHC software is only successful in the hands of colossal collaborations with round-the-clock operations teams and on-call experts.

Funding agencies are aware that they cannot fund software development in each small experiment at an adequate level unless there is a way to support these experiments with cross-cutting software. In my opinion, we, the HEP community have to show how this can be done.

4.6. Mountains to climb

HEP has internal and external issues. Paralleling the well-founded distrust of management wisdom in HEP software, is the wide distrust of the word “collaborate.” When a major institute says “collaborate with us to develop software” it is generally interpreted as “we lead, you follow.”

From outside our field we are often viewed as arrogant or untrustworthy. When we describe the genuine success of our software for the LHC (or BaBar or …), the reaction of a non-HEP scientist is often “sorry, you have great hammers but we have no nails.” Our insistence that this is not true is treated as arrogance, at least in my personal experience.

When we propose collaborative projects with other sciences, we are often perceived to be after their funding rather than their collaboration. Again this is from personal experience – for example a private conversation with an associate director of a funding agency.

None of these mountains is impossible to climb, but if we ignore their existence we will significantly postpone our success.

5. Towards an HEP Software Foundation

There are two clear imperatives driving HEP to improve the coherence of its software development. Firstly we must make the best possible use of the resources we have and secondly we must position ourselves to be attractive targets for non-HEP funding. These imperatives were certainly in the minds of CERN management in supporting a first meeting on HEP Software Collaboration[8] at CERN on April 3 and 4, 2014. This meeting was well attended and lively, involving several free and frank exchanges of views! The outcome was an agreement to solicit brief white papers on what an HEP Software Foundation (HSF) should do, and how it should be governed. I believe that "Collaboration" became "Foundation" because the latter did not have the negative connotations that I mention above.

Ten white papers[9] were received, and to the disappointment of those expecting more free and frank discussion, there was remarkable agreement on what an HSF should do, and that it should start to do it in a "bottom-up" way with vaguely defined, or even truly minimal governance.

The embryonic HSF is now in existence. It is governed by an "Interim Foundation Board" (iFB) whose membership is the set of people who choose to turn up at its monthly video meetings. The iFB solicited proposals for an initial "Leadership Team" and converged rapidly on a small group that agreed to get started and show that the Foundation could be both useful and non-threatening.

I will be following this new "experiment" with great interest!

6. Conclusion

Computing in high-energy physics is vitally important to our success and presents an intellectual challenge comparable to detector design or physics analysis. Our computing is also a significant part of the cost of the science, especially during the operation phase of experiments.

HEP computing presents technological and intellectual challenges that help in attracting enthusiastic and capable effort. HEP computing is also sociologically challenging – it often does not lead to recognition and career success and while not needing the disciplined management required for detector construction, it struggles to find the appropriate level of organization.

But finally, I have to admit that I have continued my involvement in HEP computing mainly because, like any challenging work with smart and interesting people, it is fun.

[8] Agenda: http://indico.cern.ch/event/297652/

[9] See http://ph-dep-sft.web.cern.ch/content/white-papers-contributed-discussion-hep-software-foundation-0

References

[1] J.J. Aubert et al. Phys. Lett. B 123B (1983) 275–278
[2] R.P. Mount, Nucl.Instrum.Meth. 187 (1981) 401-405.
[3] Dwight Eisenhower, from a speech to the National Defense Executive Reserve Conference in Washington, D.C. (November 14, 1957) ; in *Public Papers of the Presidents of the United States, Dwight D. Eisenhower, 1957*, National Archives and Records Service, Government Printing Office, p. 818 : ISBN 0160588510, 9780160588518
[4] Proceedings of the CERN School of Computing - CSC '86, https://cds.cern.ch/record/178680/files/CERN-87-04.pdf, page 20.
[5] T. Berners-Lee and R. Cailliau, Proposal for a hypertext project, http://www.w3.org/Proposal.html
[6] IBM Model 0663 specification, http://www.hgst.com/tech/techlib.nsf/techdocs/d49c63ecd1d8207487256c62006b797a/$file/0663-e1x_ps.pdf
[7] G. Moore, Electronics Magazine April 19, 1965 (available from http://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf
[8] S. Jarp et al. Journal of Physics: Conference Series 331 (2011) 052030.
[9] C. Mellor, The Register, January 14, 2014, www.theregister.co.uk
[10] http://www.telegeography.com/research-services/global-bandwidth-research-service/
[11] D. O. Williams, CERN Academic Training Lectures, 2005 http://indico.cern.ch/event/a043645/material/5/0.pdf Slide 3 of 180.
[12] Root, a data analysis framework, root.cern.ch
[13] Xrootd, high performance, scalable fault tolerant access to data, xrootd.org
[14] Geant4, a toolkit for the simulation of the passage of particles through matter, geant4.cern.ch
[15] I. Foster and C. Kesselman (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.
[16] D. Adams, The Hitchhiker's Guide to the Galaxy, Original Radio Scripts. edited by Geoffrey Perkins. Pan Books, London. 1985. ISBN 0-330-29288-9